

Freeform Search

Database: US Pre-Grant Publication Full-Text Database
 US Patents Full-Text Database
 US OCR Full-Text Database
 EPO Abstracts Database
 JPO Abstracts Database
 Derwent World Patents Index
 IBM Technical Disclosure Bulletins

Term: @pd<20030623 and L256

Display: 10 **Documents in Display Format:** - **Starting with Number** 1

Generate: ☐ Hit List ☒ Hit Count ☐ Side by Side ☐ Image

Search

Clear

Interrupt

Search History

DATE: Monday, April 16, 2007 [Purge Queries](#) [Printable Copy](#) [Create Case](#)

<u>Set Name</u> side by side	<u>Query</u>	<u>Hit Count</u>	<u>Set Name</u> result set
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR		
L257	@pd<20030623 and L256	21	L257
L256	l70 and L255	34	L256
L255	access\$3 near ((shared or common) adj resource)	2938	L255
L254	L247 and L253	29	L254
L253	((global or local) near (pointer or variable)) with resource	172	L253
L252	@pd<20030623 and L251	23	L252
L251	L182 and L250	44	L251
L250	L247 and L249	191	L250
L249	(variable or pointer or address) near resource	3309	L249
L248	L246 and L247	15	L248
L247	access\$3 near shared near (resource or memory)	5345	L247
L246	((thread or process or application) near (variable or pointer or address)) with resource	359	L246
L245	((exchan\$3 or switch\$3) near (variable or pointer or address)) with resource	54	L245
L244	((exchan\$3 or switch\$3) near (variable or pointer)) with resource	6	L244

<u>L243</u> ((shared near (memory or resource)) and (protect\$3 or exclus\$3)).ab,ti. <i>DB=TDBD,DWPI,JPAB,EPAB,USOC,USPT,PGPB; PLUR=YES; OP=OR</i>	260	<u>L243</u>
<u>L242</u> GOMES-CARLOS-P!	4	<u>L242</u>
<u>L241</u> GOMES-CARLOS!	1	<u>L241</u>
<u>L240</u> CHAN-HON-KEAT!	2	<u>L240</u>
<u>L239</u> CHAN-HON-KEAT-W! <i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR</i>	9	<u>L239</u>
<u>L238</u> L233 and L237	15	<u>L238</u>
<u>L237</u> thread and (heap or (shared near (resource or memory or address)))	7943	<u>L237</u>
<u>L236</u> thread and (heap or (shared near (resource or memory or address)))and L234	2155	<u>L236</u>
<u>L235</u> L233 and L234	3	<u>L235</u>
<u>L234</u> thread with (heap or (shared near (resource or memory or address)))	2155	<u>L234</u>
<u>L233</u> ((switch\$3 or exchang\$3) near variable)	8025	<u>L233</u>
<u>L232</u> @pd<20030623 and L231	75	<u>L232</u>
<u>L231</u> thread and L229	195	<u>L231</u>
<u>L230</u> thead and L229	0	<u>L230</u>
<u>L229</u> L224 and L228	354	<u>L229</u>
<u>L228</u> wait\$3 and L227	673	<u>L228</u>
<u>L227</u> return\$3 near pointer with (resource or address or memory or area or region)	1152	<u>L227</u>
<u>L226</u> @pd<20030623 and L225	61	<u>L226</u>
<u>L225</u> L223 and L224	97	<u>L225</u>
<u>L224</u> (shared or common) near (resource or address or memory or area or region)	81273	<u>L224</u>
<u>L223</u> return\$3 near pointer with (local or global or shared)	180	<u>L223</u>
<u>L222</u> L215 and L221	2	<u>L222</u>
<u>L221</u> exchang\$3 near pointer	191	<u>L221</u>
<u>L220</u> exchang\$3 with wait\$3 near pointer	1	<u>L220</u>
<u>L219</u> exchang\$3 with local adj pointer	4	<u>L219</u>
<u>L218</u> exchang\$3 near local adj pointer	0	<u>L218</u>
<u>L217</u> L215 and L216	34	<u>L217</u>
<u>L216</u> pointer near (resource or memory or area or address or space or region)	21402	<u>L216</u>
<u>L215</u> pointer near wait\$3	153	<u>L215</u>
<u>L214</u> (thread or task) and L213	63	<u>L214</u>
<u>L213</u> pointer adj wait\$3	94	<u>L213</u>
<u>L212</u> pointer near wait\$3 near (object or resource or component)	3	<u>L212</u>
<u>L211</u> wait\$3 near pointer	153	<u>L211</u>
<u>L210</u> wait\$3 near (pointer or variable)	557	<u>L210</u>
<u>L209</u> local near (pointer or variable) near (thread or process)	145	<u>L209</u>
<u>L208</u> thread near (swap\$4 or switch\$3 or exchang\$3) near (pointer or address)	25	<u>L208</u>
<u>L207</u> L205 and L206 (global or shared) near (pointer or variable) with ((shared or common or secure or protect\$3) near (object or resource or memory or area or address or space or region))	23	<u>L207</u>
	392	<u>L206</u>

<u>L205</u>	local near (pointer or variable) with (thread or process)	817	<u>L205</u>
<u>L204</u>	L200 and L203	31	<u>L204</u>
<u>L203</u>	(exchang\$3 or switch\$3 or swap\$4 or chang\$3 or replac\$6) near pointer	5657	<u>L203</u>
<u>L202</u>	L200 and L201	2	<u>L202</u>
<u>L201</u>	pointer near wait\$3	153	<u>L201</u>
<u>L200</u>	pointer near ((shared or common or secure or protect\$3) near (object or resource or memory or area or address or space or region))	277	<u>L200</u>
<u>L199</u>	pointer near ((shared or common or secure or protect\$3) near (resource or memory or area or address or space or region))	244	<u>L199</u>
<u>L198</u>	5057996.pn.	2	<u>L198</u>
<u>L197</u>	(swap\$4 or switch\$3 or exchang\$3) near (pointer or address) with ((shared or common or secure or protect\$3) near (resource or memory or area or address or space or region))	105	<u>L197</u>
<u>L196</u>	indirect near pointer with (swap\$4 or switch\$3 or exchang\$3)	4	<u>L196</u>
<u>L195</u>	indirect near pointer with (swap\$4 or switch\$3 or exchang\$3) near (function or procedure or instruction)	0	<u>L195</u>
<u>L194</u>	(local with (global or shared) near pointer)	75	<u>L194</u>
<u>L193</u>	L45 and L192	19	<u>L193</u>
<u>L192</u>	(swap\$4 or switch\$3 or exchang\$3) near (function or procedure or instruction) near address	185	<u>L192</u>
<u>L191</u>	chang\$3 near local near pointer	8	<u>L191</u>
<u>L190</u>	((switch\$3 or exchang\$3) with ((local or global) near pointer))	31	<u>L190</u>
<u>L189</u>	L182 and L188	44	<u>L189</u>
<u>L188</u>	L162 and L176	311	<u>L188</u>
<u>L187</u>	L182 and L185	8	<u>L187</u>
<u>L186</u>	L182 and L185	13628	<u>L186</u>
<u>L185</u>	jump\$3 with ((local or global) near variable)	67	<u>L185</u>
<u>L184</u>	L172 and L162	11	<u>L184</u>
<u>L183</u>	L177 and L182	15	<u>L183</u>
<u>L182</u>	L178 or L179 or L180 or L181	10123	<u>L182</u>
<u>L181</u>	(719/312).ccls.	188	<u>L181</u>
<u>L180</u>	(709/213 709/214 709/215 709/216).ccls.	2176	<u>L180</u>
<u>L179</u>	(711/147 711/148 711/149).ccls.	1928	<u>L179</u>
<u>L178</u>	(718/100 718/101 718/102 718/103 718/104 718/105 718/106 718/107 718/108).ccls.	6295	<u>L178</u>
<u>L177</u>	L172 and L176	142	<u>L177</u>
<u>L176</u>	(common or shared) near (resource or address or region or memory)	68224	<u>L176</u>
	<i>DB=TDBD,DWPI,JPAB,EPAB,USOC,USPT,PGPB; PLUR=YES; OP=OR</i>		
<u>L175</u>	GOMES-CARLOS!	1	<u>L175</u>
<u>L174</u>	CHAN-HON-KEAT-W!	9	<u>L174</u>
	<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR</i>		
<u>L173</u>	L171 and L172	11	<u>L173</u>
<u>L172</u>	((switch\$3 or exchang\$3) near pointer)	1212	<u>L172</u>

<u>L171</u> pointer near wait\$3	153	<u>L171</u>
<i>DB=USPT; PLUR=YES; OP=OR</i>		
<u>L170</u> '4543627'.pn.	1	<u>L170</u>
<u>L169</u> '4744023'.pn.	1	<u>L169</u>
<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR</i>		
<u>L168</u> ((switch\$3 or exchang\$3) near pointer) with (atomic\$4 or interrupt\$3)	19	<u>L168</u>
<u>L167</u> chang\$3 near pointer with (local or global)	38	<u>L167</u>
<u>L166</u> chang\$3 near-local near pointer with global	2	<u>L166</u>
<i>DB=USPT; PLUR=YES; OP=OR</i>		
<u>L165</u> '5950505'.pn.	1	<u>L165</u>
<u>L164</u> '5892899'.pn.	1	<u>L164</u>
<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR</i>		
<u>L163</u> jump\$3 near address and L162	54	<u>L163</u>
<u>L162</u> (local and global) near (pointer or variable)	1139	<u>L162</u>
<i>DB=USPT; PLUR=YES; OP=OR</i>		
<u>L161</u> '5504930'.pn.	1	<u>L161</u>
<u>L160</u> '5511207'.pn.	1	<u>L160</u>
<u>L159</u> '5392207'.pn.	1	<u>L159</u>
<u>L158</u> '5619409'.pn.	1	<u>L158</u>
<u>L157</u> '5619409'.pn.	1	<u>L157</u>
<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR</i>		
<u>L156</u> ((switch\$3 or exchang\$3) near (pointer or variable)) with ((shared or common or secure or protect\$3) near (resource or memory or area or address or space))	28	<u>L156</u>
<u>L155</u> L152 and L154	8	<u>L155</u>
<u>L154</u> (exchang\$3 or replac\$6) near pointer	1404	<u>L154</u>
<u>L153</u> L151 and L152	8	<u>L153</u>
<u>L152</u> access near (shared or common) near resource	2780	<u>L152</u>
<u>L151</u> pointer near wait\$3	153	<u>L151</u>
<u>L150</u> thread with access near resource and L149	9	<u>L150</u>
<u>L149</u> pointer near (local\$3 or global\$3)	1216	<u>L149</u>
<u>L148</u> L146 and L147	123	<u>L148</u>
<u>L147</u> shared near (area or address or memory or resource)	39353	<u>L147</u>
<u>L146</u> (replac\$6 or switch\$3 or exchang\$3 or chang\$3) with ((global or local) near (pointer or variable))	898	<u>L146</u>
<u>L145</u> wait\$3 near flag with (shared near (area or address or memory or resource))	3	<u>L145</u>
<u>L144</u> interchangeable\$3 near pointer	19	<u>L144</u>
<u>L143</u> L137 and L142	5	<u>L143</u>
<u>L142</u> exclus\$3 near access with (shared near (area or address or memory or resource))	347	<u>L142</u>
<u>L141</u> return\$ near (variable or pointer) with (shared near (area or address or memory or resource))	20	<u>L141</u>
<u>L140</u> L136 and L139	1	<u>L140</u>

<u>L139</u>	pointer near wait\$3	153	<u>L139</u>
<u>L138</u>	L136 and L137	24	<u>L138</u>
<u>L137</u>	((replac\$6 or switch\$3 or exchang\$3 or chang\$3) near pointer)	5476	<u>L137</u>
<u>L136</u>	pointer near (shared near (area or address or memory or resource))	178	<u>L136</u>
<u>L135</u>	initializ\$5 near pointer with (shared near (area or address or memory or resource))	3	<u>L135</u>
<u>L134</u>	replac\$6 near pointer with (shared near (area or address or memory or resource))	1	<u>L134</u>
<u>L133</u>	indirect near referenc\$3 with (shared near (area or address or memory or resource))	1	<u>L133</u>
<u>L132</u>	indriect near referenc\$3 with (shared near (area or address or memory or resource))	0	<u>L132</u>
<u>L131</u>	((switch\$3 or exchang\$3 or chang\$3) near pointer) with (shared near (area or address or memory or resource))	8	<u>L131</u>
<u>L130</u>	indirect near pointer with (shared near (area or address or memory or resource))	4	<u>L130</u>
<u>L129</u>	L124 and L127	36	<u>L129</u>
<u>L128</u>	L112 and L127	104	<u>L128</u>
<u>L127</u>	initializ\$4 near (global or local) near (pointer or variable)	298	<u>L127</u>
<u>L126</u>	L124 and L125	115	<u>L126</u>
<u>L125</u>	return near pointer with (area or address or memory or resource)	806	<u>L125</u>
<u>L124</u>	(exchang\$3 or chang\$3 or replac\$6) near (pointer or variable)	29600	<u>L124</u>
<u>L123</u>	L120 and L122	1	<u>L123</u>
<u>L122</u>	branch\$3 near (pointer or address) with access\$3 near (area or address or memory or resource)	145	<u>L122</u>
<u>L121</u>	L119 and L120	103	<u>L121</u>
<u>L120</u>	(exchang\$3 or chang\$3 or replac\$6) near pointer	4564	<u>L120</u>
<u>L119</u>	@pd<20030623 and L118	1637	<u>L119</u>
<u>L118</u>	pointer and L117	2260	<u>L118</u>
<u>L117</u>	L112 and L116	3753	<u>L117</u>
<u>L116</u>	branch\$3 near (pointer or address)	6984	<u>L116</u>
<u>L115</u>	@pd<20030623 and L114	181	<u>L115</u>
<u>L114</u>	pointer and L113	327	<u>L114</u>
<u>L113</u>	L111 and L112	850	<u>L113</u>
<u>L112</u>	access\$3 near (area or address or memory or resource)	409450	<u>L112</u>
<u>L111</u>	((thread or process) near (jump\$3 or thunk\$3))	3183	<u>L111</u>
<u>L110</u>	((jump\$3 or thunk\$3) with (local near variable))	27	<u>L110</u>
<u>L109</u>	((jump\$3 or thunk\$3) with (enter\$4 near (area or address or memory or resource)))	90	<u>L109</u>
<u>L108</u>	L79 and L107	56	<u>L108</u>
<u>L107</u>	(swap\$4 or switch\$3 or exchang\$3) with ((local or global or shared) near (variable or mutexes or semaphore))	254	<u>L107</u>

DB=PGPB; PLUR=YES; OP=OR

<u>L106</u>	US-20050060416-A1.did.	1	<u>L106</u>
	<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR</i>		
<u>L105</u>	(swap\$4 or switch\$3 or exchange\$3) with ((local or global or shared) near (pointer))	54	<u>L105</u>
<u>L104</u>	(swap\$4 or switch\$3 or exchange\$3) near ((local or global or shared) near (pointer))	3	<u>L104</u>
<u>L103</u>	((swap\$4 or switch\$3 or exchange\$3) near (pointer)) with wait\$3	7	<u>L103</u>
	<i>DB=USPT; PLUR=YES; OP=OR</i>		
<u>L102</u>	'5805900'.pn.	1	<u>L102</u>
<u>L101</u>	'6052695'.pn.	1	<u>L101</u>
<u>L100</u>	'6105085'.pn.	1	<u>L100</u>
<u>L99</u>	'6105085'.pn.	1	<u>L99</u>
	<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR</i>		
<u>L98</u>	L74 and L97	59	<u>L98</u>
<u>L97</u>	((swap\$4 or switch\$3 or exchange\$3) near (pointer))	1418	<u>L97</u>
<u>L96</u>	L79 and L95	35	<u>L96</u>
<u>L95</u>	((local and global) with (mutex or semaphore))	56	<u>L95</u>
<u>L94</u>	((local and global) near (mutex or semaphore))	13	<u>L94</u>
<u>L93</u>	L88 and L92	12	<u>L93</u>
<u>L92</u>	content\$5 near (shared or common) near (resource or address or memory or area or region)	689	<u>L92</u>
<u>L91</u>	L74 and L90	2	<u>L91</u>
<u>L90</u>	L88 and L81	12	<u>L90</u>
<u>L89</u>	L87 and L88	4	<u>L89</u>
<u>L88</u>	pointer near (shared or common) near (resource or address or memory or area or region)	170	<u>L88</u>
<u>L87</u>	((switch\$3 or exchange\$3) near (function or procedure or method)) with (shared or common) near (resource or address or memory or area or region)	154	<u>L87</u>
<u>L86</u>	L74 and L85	67	<u>L86</u>
<u>L85</u>	L81 and L84	584	<u>L85</u>
<u>L84</u>	(thread or process) near wait\$3	10253	<u>L84</u>
<u>L83</u>	(thread or process) near wait	7139	<u>L83</u>
<u>L82</u>	L74 and L81	1050	<u>L82</u>
<u>L81</u>	((switch\$3 or exchange\$3) near (function or procedure or method))	126303	<u>L81</u>
<u>L80</u>	L78 and L79	8	<u>L80</u>
<u>L79</u>	(shared or common) near (resource or address or memory or area or region)	81273	<u>L79</u>
<u>L78</u>	((switch\$3 or exchange\$3) near (function or procedure or method)) with (atomic\$4 or without near interrupt\$3)	134	<u>L78</u>
<u>L77</u>	L74 and L76	3	<u>L77</u>
<u>L76</u>	((switch\$3 or exchange\$3 or change\$3) near (function or procedure or method)) with (atomic\$4 or without near interrupt\$3)	243	<u>L76</u>
<u>L75</u>	L73 and L74	98	<u>L75</u>
	access\$3 near (resource or ((shared or common) near (address or memory or		

<u>L74</u>	area or region)))	21513	<u>L74</u>
<u>L73</u>	((return\$3 or relinquish\$3) near pointer) with (resource or memory)	639	<u>L73</u>
<u>L72</u>	(pointer near (resource or wait\$3))	668	<u>L72</u>
<u>L71</u>	L53 and L70	73	<u>L71</u>
<u>L70</u>	(local and (shared or global)) near (pointer or variable)	1286	<u>L70</u>
<u>L69</u>	L55 and L67	0	<u>L69</u>
<u>L68</u>	L53 and L67	30	<u>L68</u>
<u>L67</u>	(alter\$3 or chang\$3) near (local and (shared or global))	292	<u>L67</u>
<u>L66</u>	(alter\$3 or chang\$3) with (local and (shared or global))	3557	<u>L66</u>
<u>L65</u>	(alter\$3 or chang\$3) with (local and (shared or global)) near pointer	2	<u>L65</u>
<u>L64</u>	(swap\$4 or switch\$3 or exchang\$3) with (local and (shared or global)) near pointer	3	<u>L64</u>
<u>L63</u>	718/???ccls. and L61	6	<u>L63</u>
<u>L62</u>	L55 and L61	5	<u>L62</u>
<u>L61</u>	@pd<20030623 and L60	205	<u>L61</u>
<u>L60</u>	L53 and L59	306	<u>L60</u>
<u>L59</u>	(swap\$4 or switch\$3 or exchang\$3) near (local or shared or global)	18403	<u>L59</u>
<u>L58</u>	pointer and L50	25	<u>L58</u>
<u>L57</u>	pointer and L55	1418	<u>L57</u>
<u>L56</u>	L50 and L55	1	<u>L56</u>
<u>L55</u>	(swap\$4 or switch\$3 or exchang\$3) near pointer	1418	<u>L55</u>
<u>L54</u>	L50 and L53	6	<u>L54</u>
<u>L53</u>	(access\$3) near ((shared or common or protected) near (resource or address or region or portion or memory or area))	9240	<u>L53</u>
<u>L52</u>	L50 and L51	1	<u>L52</u>
<u>L51</u>	(pointer) near ((shared or common or protected) near (resource or address or region or portion or memory or area))	235	<u>L51</u>
<u>L50</u>	(branch\$3 or jump\$3) near ((shared or common or protected) near (resource or address or region or portion or memory or area))	161	<u>L50</u>
<u>L49</u>	(branch\$3 or jump\$3) near ((shared or common or protected) near (address or region or portion or memory or area))	161	<u>L49</u>
<u>L48</u>	L45 and L47	21	<u>L48</u>
<u>L47</u>	(branch\$3 or jump\$3) near (swap\$4 or switch\$3 or exchang\$3) near (function or procedure or instruction)	253	<u>L47</u>
<u>L46</u>	L44 and L45	46	<u>L46</u>
<u>L45</u>	((shared or common or protected) near (address or region or portion or memory or area))	114146	<u>L45</u>
<u>L44</u>	(swap\$4 or switch\$3 or exchang\$3) near (function or procedure or instruction) with ((memory or address) near (location or pointer))	153	<u>L44</u>
<u>L43</u>	(swap\$4 or switch\$3 or exchang\$3) near (function or procedure or instruction) near pointer	32	<u>L43</u>
<u>L42</u>	L37 and L41	90	<u>L42</u>
<u>L41</u>	(pointer or address) near wait\$3	1126	<u>L41</u>

<u>L40</u>	(pointer or address) near wait	687	<u>L40</u>
<u>L39</u>	L37 and L38	52	<u>L39</u>
<u>L38</u>	(exchang\$3 near (variable or pointer or address))	3016	<u>L38</u>
<u>L37</u>	(pointer or address) near ((shared or common) near (resources or address or region or memory or space) or heap)	7003	<u>L37</u>
<u>L36</u>	thread and L35	17	<u>L36</u>
<u>L35</u>	L33 and L34	133	<u>L35</u>
<u>L34</u>	(shared or common) near (address or region or memory or space) or heap	75352	<u>L34</u>
<u>L33</u>	(exchang\$3 or switch\$3) near (local or global or shared) with (variable or pointer or address)	914	<u>L33</u>
<u>L32</u>	(exchang\$3 or switch\$3) near (variable or pointer or address) with heap	4	<u>L32</u>
<u>L31</u>	L30 and L28	42	<u>L31</u>
<u>L30</u>	(exchang\$3 or switch\$3) near (tasks or thread)	6072	<u>L30</u>
<u>L29</u>	switch\$3 near (tasks or thread)	5312	<u>L29</u>
<u>L28</u>	global adj variables and local adj variables	916	<u>L28</u>
<u>L27</u>	thread and L26	54	<u>L27</u>
<u>L26</u>	@pd<20030623 and L25	962	<u>L26</u>
<u>L25</u>	L21 and L24	1522	<u>L25</u>
<u>L24</u>	(exchang\$3 or switch\$3) near (pointer or address or variable)	20641	<u>L24</u>
<u>L23</u>	L21 and L22	1420	<u>L23</u>
<u>L22</u>	(exchang\$3 or switch\$3) near (pointer or address)	12703	<u>L22</u>
<u>L21</u>	(shared or common) near (address or region or memory or space)	61156	<u>L21</u>
<u>L20</u>	L14 and L19	47	<u>L20</u>
<u>L19</u>	thread with (heap or (shared near (resource or memory)))	2135	<u>L19</u>
<u>L18</u>	L14 and L17	1	<u>L18</u>
<u>L17</u>	thread near (heap or (shared near (resource or memory)))	230	<u>L17</u>
<u>L16</u>	thread and L15	117	<u>L16</u>
<u>L15</u>	L14 and (heap or (shared near (resource or memory)))	188	<u>L15</u>
<u>L14</u>	class near pointer	847	<u>L14</u>
<u>L13</u>	class with (property near pointer) and (heap or (shared near (resource or memory)))	4	<u>L13</u>
<u>L12</u>	((exchang\$3 or switch\$3) near (pointer or address)) with (heap or (shared near (resource or memory)))	16	<u>L12</u>
<u>L11</u>	L9 and L10	38	<u>L11</u>
<u>L10</u>	pointer near class	847	<u>L10</u>
<u>L9</u>	pointer near (heap or (shared near (resource or memory)))	403	<u>L9</u>
<u>L8</u>	thread near pointer with (heap or (shared near (resource or memory)))	2	<u>L8</u>
<u>L7</u>	thread near pointer near (function or method)	8	<u>L7</u>
<u>L6</u>	L4 and L5	27	<u>L6</u>
<u>L5</u>	request with (heap or (shared near (resource or memory)))	4359	<u>L5</u>
<u>L4</u>	thread with (local near (address or pointer or reference or variable))	400	<u>L4</u>
<u>L3</u>	(local and global) near (shared near resource)	3	<u>L3</u>

L2 ((local or global) near (pointer or variable or address)) with (shared near resource)

25 L2

L1 20030061394

3 L1

END OF SEARCH HISTORY

Find:

Searching for **PHRASE** exchange variable shared resource.

Restrict to: [Header](#) [Title](#) Order by: [Expected citations](#) [Hubs](#) [Usage](#) [Date](#) Try: [Google \(CiteSeer\)](#)
[Google \(Web\)](#) [Yahoo!](#) [MSN](#) [CSB](#) [DBLP](#)

No documents match Boolean query. Trying non-Boolean relevance query.

500 documents found. Only retrieving 250 documents (System busy - maximum reduced). Order: relevance to query.

[Resource Sharing in Reservation-Based Systems - de Niz., Saewong, Rajkumar.. \(2001\) \(Correct\) \(4 citations\)](#)

Resource Sharing in Reservation-Based Systems Dionisio de Niz,

Resource Sharing in Reservation-Based Systems Dionisio de

real-time operating systems began to support the **resource** reservation paradigm. This technique has proved
www.cs.cmu.edu/afs/cs/project/rtml-2/Papers/RTAS01/deniz.ps.gz

[Modification of Shared Resources in AL-1/D - Hideaki Okamuray \(Correct\)](#)

sweep, reference counting and so on. We want to **exchange** these algorithms and to evaluate new GC
 line 5 to line 8. This meta object has internal **variables** such as gc, scheduler, allocator, segment,

Modification of **Shared Resources** in AL-1/D Hideaki Okamuray

jerry.cs.uiuc.edu/reflection/washington/postscript/okamura-letter.ps

[A Comparative Study of Two Generic Resource Allocation Models - Youssef Abdel-Wahab \(1997\) \(Correct\)](#)

arise in distributed systems is the allocation of **shared resources** among a group of clients. This problem

A Comparative Study of Two Generic **Resource** Allocation Models A. Youssef, H. Abdel-Wahab,

Norfolk, VA 23529 Abstract In a typical **resource** allocation model, clients request a fixed amount

www.cs.odu.edu/~youssef/papers/NC97.ps

[A New Distributed Resource-Allocation Algorithm with.. - Sivilotti, Pike, Sridhar \(2000\) \(Correct\) \(2 citations\)](#)

problem [6]Although rst postulated as a **shared**-memory concurrency problem, it has since received

A New Distributed **Resource**-Allocation Algorithm with Optimal Failure

problem {a classic problem in distributed **resource** allocation {that has optimal failure locality.

www.cis.ohio-state.edu/~nsridhar/Research/Papers/PS/fl.ps

[Mutual Exclusion on Optical Buses - Kavi, Mehta \(Correct\)](#)

contain the same values in each processor. The **variable** my turn[r] for a **resource** r in processor P i

survey)To the best of our knowledge, the **shared**-memory paradigm for parallel computing has not

it arises whenever concurrent access to **shared resources** by several sites is involved [4]For

crash1.eb.uah.edu/~kavi/Research/pdcs-00-mutex.pdf

[Search Tree Unification: Paradigm for Process-based Logic.. - Ferenczi \(Correct\)](#)

with the same communication pattern and unifying **variables** used in the individual communications [FF92] as

Recent blackboard oriented languages like **Shared** Prolog [BC91] belong to process-based languages

the program. This partition serves as **shared resource** for the processes, just like blackboards in

ftp.elis.rug.ac.be/pub/prolog/iclp94_workshop/ferenczi.ps.Z

[The Mutual Exclusion Problem Has Been Solved - Lamport \(1990\) \(Correct\)](#)

underlying mutual exclusion algorithms in **shared** memory systems is that: b. A memory reference to

executing processes do not access a **shared resource** at the same time. In a **shared** memory system,

memory system, each word of memory is a **shared resource**. Assumption b states that two concurrently

research.compaq.com/SRC/personal/lamport/pubs/lamport-mutual-solved.ps.Z

[A Survey of Distributed Mutual Exclusion Algorithms - Velazquez \(1993\) \(Correct\) \(3 citations\)](#)

algorithms, failures in the system need the **exchange** of more information than in normal operation, in

finite amount of time, but the time of arrival is **variable**. The topology of the network is known. Nodes

distributed computing systems are many. **Resource sharing**, parallel processing, system availability and

www.cs.colostate.edu/~ftppub/TechReports/1993/tr-116.ps.Z

[Local Serialization Pattern - Silva, Pereira, Sousa \(Correct\)](#)

other concurrent access modes, e.g. if a **shared variable** is being written it cannot be simultaneously

for the concurrency control concern within a **shared resource**. 1 Introduction In this paper we the concurrency control concern within a **shared resource**. 1 Introduction In this paper we present the siesta.cs.wustl.edu/~schmidt/OOPSLA-95/html/papers/atomobj.ps.gz

Verifying Mutual Exclusion and Liveness Properties with TLA - Pau, Rents, Schreiner (2000) (Correct)

In this algorithm we use the following **shared variables**: turn 2 f0 1g f lag i 2 f0 1g where i 2 the following problem: given two processes and a **shared resource**, which must not be accessed by these two problem: given two processes and a **shared resource**, which must not be accessed by these two www.risc.uni-linz.ac.at/people/schreine/papers/extla.ps.gz

Yet Another Distributed Mutual Exclusion Algorithm - Teixeira, de Andrade.. (Correct)

of a single token. The average number of messages **exchanged** per critical section is $O(\log N)$ where N is its parent. For instance, the node i maintains a **variable** parent i which indicates its parent node. two or more nodes which simultaneously request the **shared resource**. Such queue is controlled by the current www.cs.umd.edu/~hcma/papers/sccc95.ps.gz

Tight Space Self-stabilizing Uniform I-Mutual Exclusion - Gradinariu, Tixeuil (2000) (Correct)

where the processors maintain two types of **variables**: local **variables** and eld **variables**. The local exclusion :the system has to guarantee the fair **sharing** of a **resource** that can be used by l processors the system has to guarantee the fair **sharing** of a **resource** that can be used by l processors www.lri.fr/~mariag/icdcs01.ps

Fair On-Line Scheduling of a Dynamic Set of Tasks on a.. - Baruah, Gehrke, Plaxton (1996) (Correct) (12 citations)

)Otherwise, we say that the task requests are **variable**. The total request at slot t, denoted R(t) is processors, disks, network bandwidth) may be **shared** among different "tasks" e.g.user processes) Scheduling of a Dynamic Set of Tasks on a Single **Resource** Sanjoy K. Baruah Johannes E. Gehrke y C. www.cs.wisc.edu/~johannes/papers/tr96-03.ps

Non-Preemptive Real-Time Scheduling Of Dataflow Systems - Parks, Lee (1995) (Correct)

enabled subtasks. An activation frame holds local **variables** that are **shared** by the subtasks. When a include tasks that contend for exclusive access to **shared resources**. We show that non-preemptive tasks that contend for exclusive access to **shared resources**. We show that non-preemptive rate-monotonic ptolemy.eecs.berkeley.edu/~parks/papers/icassp95-parks/icassp95.ps.Z

Bandwidth Regulation of Real-Time Traffic Classes in.. - Liebeherr, Akyildiz.. (1994) (Correct) (2 citations)

Rate p (gw:li) and OL p (gw:li) and two **variables**, **Share** p (gw:li) and **Surplus** p (gw:li) for has identical bandwidth constraints if their routes **share** the link with the smallest capacity for this solution is appropriate if sufficient network **resources** are always available. Additionally, one may ftp.cs.virginia.edu/pub/jorg/papers/CS-94-24.ps

Some Performance Issues Associated with CEWES MSRC Scalable.. - Bova (1997) (Correct)

product requires point-to-point communication for **exchange** of data on the processor interfaces, and the dot Computing Modernization Program CEWES Major **Shared Resource** Center through Programming Environment Modernization Program CEWES Major **Shared Resource** Center through Programming Environment and www.wes.hpc.mil/pet/tech_reports/reports/pdf/tr_9702.pdf

The ccNUMA Memory Profiler - Alan Davis Ald (Correct)

increasingly popular. Compared to the traditional **shared-bus** (SMP) systems, they scale to much higher scalability is not limited by a single **shared resource**. Compared to other scalable multiprocessor ccNUMA designs replace one **shared resource** (system bus) with a collection of distributed www.cs.utah.edu/~ald/pubs/CC-numa.pdf

First 20 documents [Next 20](#)

Try your query at: [Google \(CiteSeer\)](#) [Google \(Web\)](#) [Yahoo!](#) [MSN](#) [CSB](#) [DBLP](#)

CiteSeer.IST - Copyright [Penn State](#) and [NEC](#)

IP.com
PriorArtDatabase

April 16, 2007

USPTO

Secure

Search

Full Text

Concept

Document ID

Recent Disclosures

Other

Prior Art Home

Support

Logout

Displaying records #491 through 500 out of 500
(search stopped at 500 hits)

Result # 491 Relevance:

Lock-less Concurrent Dual Chaining of Control Structures

2005-07-18

IPCOM000126443D

En

The design required being able to chain a single element on two independent linked-list however there was an additional requirement that NO lock be obtained to perform the management. The solution involves creating new control structures ...

Result # 492 Relevance:

Common Language-Independent Interface for Culturally Sensitive Fun

1994-07-01

IPCOM000113046D

En

The disclosed method allows common interfaces for culturally sensitive functions to be application internationalization. These can be used across operating platforms and prog languages to support multiple cultures without the need to change the operation ...

Result # 493 Relevance:

Overlapped Field Processing

1972-06-01

IPCOM000077173D

En

Certain programming languages allow data structures to be defined in such a manner th characters or bits may be subdivided into overlapping subfields or substrings. During th compilation of a program written in such languages, it is first necessary ...

Result # 494 Relevance:

Floating Dialog Window for Presentation Manager

1991-10-01

IPCOM000121964D

En

This article discusses a technique to display window information using the mouse pointer information can be obtained via mouse pointer position and displayed in an information information window is positioned opposite the mouse pointer, all visible ...

Result # 495 Relevance:

Simple Cryptographic Program Interface (Crypto API) (RFC2628)

1999-06-01

IPCOM000003215D

En

This document describes a simple Application Program Interface to cryptographic functi purpose of such an interface is to separate cryptographic libraries from internet applica allowing an independent development of both. It can be used in various ...

Result # 496 Relevance:

Method of Configuring a Dynamic Mouse Pointer

1993-06-01

IPCOM000105211D

En

Prior art allows certain information to be displayed dynamically in the window title bar concatenation of the window title and pre-configured information. This prior art include information such as incoming mail status or a search results list into the ...

Result # 497 Relevance:

Module to Module Linkage Mechanism

1982-12-01

IPCOM000050907D

En

This article presents definition and construction of resource manager to resource manager functions. Reference to application to resource manager linkage services is made only when the service function is common to both types of linkage.

Result # 498 Relevance: 

Hypertext Capability for Interactive Edit of Program Source

1997-05-01

IPCOM000118699D

En

Disclosed is a mechanism for introducing a hypertext capability for processing program during an interactive program edit.

Result # 499 Relevance: 

Object selection method by dynamically changing units of selection

2002-10-10

IPCOM000015942D

En

Disclosed is a solution for an object selection method for a software application. In a traditional software application, the common way for object selection using a mouse requires deciding starting and ending points and placing the mouse pointer at both points ...

Result # 500 Relevance: 

A whole program register promotion method for global variables

2003-05-21

IPCOM000012694D

En

This is a method to allocate registers to global variables through the entire program.

Displaying page 50 of 50 << FIRST | < BACK | NEXT > | LAST >>

Search query: exchange function comprising changing a local pointer for a global pointer

[New search](#) | [Modify this search](#)

Copyright © 2007 IP.com, Inc. All rights reserved.

[Web](#) [Images](#) [Video](#) [News](#) [Maps](#) [more »](#)

shared resource "exchanging variable "

Search Patents

[Advanced Patent Search](#)
[Google Patent Search](#)

Patents

Patents 1 - 2 on shared resource "exchanging variable ". (0.13 seconds)

Method and apparatus for exchanging variable length frames by fixed length cell handling exchange

US Pat. 5610918 - Filed Jan 31, 1994 - Fujitsu Limited

Also where the **resource** of the ATM exchange 200 is **shared** commonly by the ATM terminals and the frame relay terminals, control of the ATM terminal ...

Multiprocessor system having statically determining resource allocation schedule at compile time and the using of static schedule with processor signals to control the execution time dynamically

US Pat. 5367678 - Filed Dec 6, 1990 - The Regents of the University of California

The STATICALLY DETERMINING **RESOURCE** other processor reads the ... particular data set is ready for **shared** memory facilitates **exchanging variable** amounts the ...

shared resource "exchanging variable "

Search Patents

[Google Patent Search Help](#) | [Advanced Patent Search](#)[Google Home](#) - [About Google](#) - [About Google Patent Search](#)

©2007 Google

[Web](#) [Images](#) [Video](#) [News](#) [Maps](#) [more »](#)[Advanced Patent Search](#)
[Google Patent Search](#)

Patents

Your search - **shared resource "exchanging pointer"** - did not match any documents.

Suggestions:

- Make sure all words are spelled correctly.
- Try different keywords.
- Try more general keywords.
- Try fewer keywords.
- Try your search at www.uspto.gov

[Google Patent Search Help](#) | [Advanced Patent Search](#)[Google Home](#) - [About Google](#) - [About Google Patent Search](#)

©2007 Google